The speaker partitioning problem

Niko Brümmer and Edward de Villiers

AGNITIO, South Africa

{nbrummer|edevilliers}@agnitio.es

Abstract

We give a unification of several different speaker recognition problems in terms of the general *speaker partitioning problem*, where a set of N inputs has to be partitioned into subsets according to speaker. We show how to solve this problem in terms of a simple generative model and demonstrate performance on NIST SRE 2006 and 2008 data. Our solution yields probabilistic outputs, which we show how to evaluate with a cross-entropy criterion. Finally, we show improved accuracy of the generative model via a discriminatively trained re-calibration transformation of log-likelihoods.

1. Introduction

The canonical speaker detection problem involves deciding whether *two* given speech utterances, denoted *train* and *test*, are spoken by the same speaker or by different speakers. The usual generalization of this problem is to supply multiple training utterances, all known to be of a target speaker and then to ask whether the test is from the target or not.

The goal of this paper is to generalize further. We propose a definition of *the most general* speaker recognition problem, when $N \geq 2$ speech utterances (each from a single speaker) are given. Then we give a practical solution to this problem, which we experimentally demonstrate.

We define the most general N-input speaker recognition problem to be *the speaker partitioning problem.* In this problem it is required of the speaker recognizer to partition the set of N inputs into M subsets, where M is the recognizer's estimate of the number of speakers and where each subset should contain all of the inputs of one of the speakers. For large N, this is a difficult problem, because there is a combinatorial explosion of ways to partition a set of size N.

In the rest of this paper we discuss the partitioning problem in more detail and show how it is related to other problems that have been addressed in the literature and in the NIST Speaker Recognition Evaluations. Then we show how to implement solutions to the most general problem, as well as a few specializations, by using a state-of-the-art 'i-vector' speaker recognizer. Our solutions are tractable for small N, while problems with large N remain challenging.

We conclude with an experimental demonstration on data from NIST's 2006 and 2008 Speaker Recognition Evaluations. We experiment with our solution to the *counting problem*, which is of intermediate generality (more general than the canonical detection problem and more specific than the partitioning problem), where the recognizer has to estimate whether there are 1,2 or 3 speakers present in a set of 3 input utterances.

2. Notation

In this section we define the necessary notation to express the speaker recognition problems discussed in this paper.

The reader will possibly find the notation unorthodox. It is customary to express solutions for speaker detection problems in terms of *likelihood-ratios*. In this work however, we find it more convenient to replace likelihood-ratios with *functions that map priors to posteriors*. These functions perform the same job as the traditional likelihood-ratios, but generalize more naturally to cases where there are more than two hypotheses.

In every problem, the *input* is a *set* of $N \geq 2$ speech utterances, denoted $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$, where each utterance x_i is assumed spoken by a single speaker.

In every problem there is a set of K hypotheses, $\Theta_K = \{\theta_1, \theta_2, \dots, \theta_K\}$, of which *exactly one* is true of the input set $\mathcal{X}$, but it is not known which one of these hypotheses is true.

Let $\mathbb{P}_K$ denote the set (simplex) in which probability distributions for $\theta \in \Theta_K$ live. If $\mathbf{p} \in \mathbb{P}_K$ and $\mathbf{p} = (p_1, p_2, \dots, p_K)$, then $p_i = P(\theta_i|\mathbf{p})$ is the probability given by $\mathbf{p}$ for θ_i to be true of $\mathcal{X}$.

There is a parameter, $\boldsymbol{\pi} \in \mathbb{P}_K$, known as the *prior* and which is independent of the input and of the recognizer. If $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots, \pi_K)$, then $\pi_i = P(\theta_i|\boldsymbol{\pi})$ is the prior probability for hypothesis θ_i. In practice, the prior is supplied by the user of the speaker recognizer.

In every problem, the *solution* is required to be a *function* which maps input and prior to posterior. A solution, say $\mathcal{R}$, must have the form $\mathbf{r} = \mathcal{R}(\mathcal{X}, \boldsymbol{\pi})$, where $\mathbf{r} = (r_1, r_2, \dots, r_K) \in \mathbb{P}_K$ and where $r_i = P(\theta_i|\mathcal{X}, \boldsymbol{\pi}, \mathcal{R})$ is the *recognizer's posterior* for hypothesis θ_i. A solution enables its user to compute a posterior for any given input and prior.

A solution is considered *good*, if its posterior distributions can be used to make minimum-expected-cost Bayes decisions that have lower cost on average than Bayes decisions made with the prior alone. In our experiments, we shall apply this test to our proposed solution.

3. Catalogue of problems

In this section, we give a detailed description of the speaker partitioning problem and we show how it is related to other more specific problems known from the literature and NIST Speaker Recognition Evaluations. We present this section in the form of a catalogue of several different speaker recognition problems.

3.1. The canonical speaker detection problem

The input is a set of $N = 2$ speech utterances, $\mathcal{X} = \{x_1, x_2\}$ and there are $K = 2$ hypotheses, $\{\theta_{\text{tar}}, \theta_{\text{non}}\}$, where θ_{tar} states that inputs x_1 and x_2 are from the same speaker and θ_{non} states they are from different speakers. Traditionally [1, 2], the solution $\mathcal{R}(\mathcal{X}, \boldsymbol{\pi}) = (r_{\text{tar}}, r_{\text{non}})$ is implemented in the form:

$$\lambda = \frac{P(\mathcal{X}|\theta_{\text{tar}}, \mathcal{R})}{P(\mathcal{X}|\theta_{\text{non}}, \mathcal{R})}, \tag{1}$$

$$\begin{aligned} r_{\text{tar}} &= P(\theta_{\text{tar}}|\mathcal{X}, \boldsymbol{\pi}, \mathcal{R}) = 1 - r_{\text{non}} \\ &= \left(1 + \left(\frac{\pi_{\text{tar}}}{\pi_{\text{non}}}\lambda\right)^{-1}\right)^{-1} \end{aligned} \tag{2}$$

where λ is the *speaker detection likelihood-ratio* and where $\pi_{\text{tar}} = P(\theta_{\text{tar}}|\boldsymbol{\pi}) = 1 - \pi_{\text{non}}$ is the prior.

3.2. The speaker partitioning problem

This is the most general of the problems in our catalogue. The input is a set of N speech inputs, $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$, where $N > 2$. There is a set of B_N hypotheses, where B_N is the Nth *Bell number*, or the number of ways a set of N elements can be partitioned [3]. The first few Bell numbers are listed in Table 1.

Each hypothesis gives a different way to partition $\mathcal{X}$ into subsets $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_M$, such that each subset has utterances from only one speaker and no two subsets share a speaker. In other words, each hypothesis states the hypothesized number of speakers, M, as well as a hypothesized partitioning of the inputs into M subsets. We denote the hypotheses in the following way:

$\theta_{12\cdots N}$ is the coarsest partition, where all N inputs are hypothesized to be of the same speaker.

$\theta_{13|245|\cdots|\cdots}$ is a partition where $\{x_1, x_3\}$ has one speaker, $\{x_2, x_4, x_5\}$ has another, and so on.

$\theta_{1|2|\cdots|N}$ is the finest partition, with N hypothesized speakers.

The canonical problem is a special case of the partitioning problem, where $N = 2$ and $\theta_{\text{tar}} = \theta_{12}$ and $\theta_{\text{non}} = \theta_{1|2}$.

We denote a solution to the partitioning problem as $\mathbf{r} = \mathcal{R}(\mathcal{X}, \boldsymbol{\pi})$, where $\mathbf{r}, \boldsymbol{\pi} \in \mathbb{P}_{B_N}$. We consider the partitioning problem to be *difficult*, simply because the prior, $\boldsymbol{\pi}$, and posterior, $\mathbf{r}$, have a very large number, B_N, of components. For example, $B_{10} > 10^5$. To compute even one component, $P(\theta|\mathcal{X}, \boldsymbol{\pi}, \mathcal{R})$, of the posterior in a straightforward way requires summing the denominator over *all* of the likelihoods for each of the B_N hypotheses (see (12) below).

3.3. The triple input problem

As an example of the partitioning problem, we consider the *triple input problem*. The utterances $\mathcal{X} = \{x_1, x_2, x_3\}$ may be spoken by one, two or three speakers and there are $B_3 = 5$ *partitioning hypotheses*, each stating that the utterances are partitioned according to speaker as:

θ_{123}: 1 speaker.

$\theta_{12|3}$: 2 speakers, x_1 and x_2 are from the same speaker.

$\theta_{13|2}$: 2 speakers, x_1 and x_3 are from the same speaker.

$\theta_{1|23}$: 2 speakers, x_2 and x_3 are from the same speaker.

$\theta_{1|2|3}$: 3 speakers.

3.4. The counting problem

The speaker counting problem has N inputs, but is a simplification of the partitioning problem, because it has just N hypotheses, $\{\theta_1, \theta_2, \dots, \theta_N\}$, where θ_i hypothesizes that there are i speakers amongst the N inputs.

Solutions to the counting problem can be expressed in terms of solutions to the partitioning problem. For example, when $N = 3$, then

$$\begin{aligned} \theta_1 &= \theta_{123}, \\ \theta_2 &= \theta_{12|3} \vee \theta_{1|23} \vee \theta_{13|2}, \\ \theta_3 &= \theta_{1|2|3} \end{aligned} \tag{3}$$

where $\vee$ denotes logical or.

In general the probability (posterior or prior) for i speakers is just the sum of the probabilities for all the different partitions that have i subsets.

3.5. The extended training detection problem

The *extended training detection problem* has an input set, $\mathcal{X} = \mathcal{T} \cup \{x_t\}$, where $\mathcal{T}$ is known as the *training set* and x_t as the *test input*. The inputs in $\mathcal{T}$ are known to be of the same speaker. There are just two hypotheses:

θ_{tar}: x_t has the same speaker as the training set.

θ_{non}: x_t has a different speaker.

This problem is well represented in the literature and has been exercised in several NIST Speaker Recognition Evaluations [4].

Solutions to this problem can be expressed in terms of solutions for the partitioning problem by using a prior that assigns zero cost to all but two of the partitioning hypotheses. As an example, when $\mathcal{T} = \{x_1, x_2\}$ and $x_t = x_3$, then $\theta_{\text{tar}} = \theta_{123}$ and $\theta_{\text{non}} = \theta_{12|3}$.

3.6. The unsupervised adaptation detection problem

This problem puts a twist on extended training by relaxing the assumption that all the speakers in the training set are the same. Here, the input set is $\mathcal{X} = \{x_T\} \cup \mathcal{A} \cup \{x_t\}$, where x_T is the *training* example of the target speaker, $\mathcal{A}$ is the *adaptation* set and x_t is the *test* input. The motivation for this flavour of detection is that if the prior probability for finding the target speaker in the adaptation set is high enough, then accuracy benefits similar to those observed in extended training may be expected.

This task was prescribed by NIST in the 2006 [5] and 2008 [6] speaker recognition evaluations. However, NIST failed to specify a prior probability for finding targets in the adaptation set, which left participants at the mercy of the unpredictable proportions of targets in the evaluation data. In our opinion, the unsupervised adaptation problem can only be tackled in a principled way if more detailed prior information is given about the adaptation set.

3.7. The diarization problem

Speaker diarization [7] is the task of annotating a conversation between two (or sometimes more) speakers, recorded in a single (2-wire telephone) channel, in order to show where each speaker is speaking. It is assumed that the diarization system has no previous exposure to any of the speakers involved. The usual solution to this problem iterates these steps until convergence:

1. Segment the recording into a number, N, of speech segments, trying to avoid segments that contain more than one speaker and trying to avoid very short segments.

2. Assuming each segment has a single speaker, do *speaker partitioning*, i.e. the problem described in section 3.2.

3. Improve the segmentation, using the results of step 2.

Table 1: Bell numbers, B_N, versus the number of non-empty subsets of a set of N elements.

N	2	3	4	5	6	7	8	9	10
$2^N - 1$	3	7	15	31	63	127	255	511	1023
B_N	2	5	15	52	203	877	4140	21147	115975

4. Repeat from step 2 until convergence.

We note that the solution for speaker partitioning that we propose in section 4.2 is ill-suited for diarization, because N tends to be large in step 2 and our method becomes intractable for large N. It becomes intractable because we do an exact computation of the posterior. For a principled way of computing an approximate, but tractable, posterior for step 2 of the diarization problem, using variational Bayes, see [8, 9].

3.8. Speaker identification

Finally, in order to emphasize the generality of the partitioning problem, we note that open-set and closed-set *speaker identification* are also special cases of the partitioning problem. In these problems, multiple inputs are given, some with known speakers and others with unknown speakers. Then the recognizer has to decide which of the known speakers (if any) are present in the utterances with unknown speakers. This problem can be expressed in terms of the partitioning problem in the obvious way.

4. The i-vector solution

Here we propose a practical approach to computing the likelihoods for the partitioning hypotheses in N-input problems. These solutions are tractable for small values of N.

This approach is based on a recent innovation [9, 10, 11], where each input utterance is represented by a *single* feature vector called[1] an *i-vector*. We apply a function f, called the *i-vector extractor*, to every input x_j, so that $\boldsymbol{\phi}_j = f(x_j)$ is the associated i-vector. The set of i-vectors, obtained by processing the input set $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ is denoted $\Phi = \{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \dots, \boldsymbol{\phi}_N\}$. In our implementation, the i-vectors are 400-dimensional.

Now we ignore the fact that we know how the i-vectors were extracted and instead pretend they were generated by some generative probabilistic model $\mathcal{M}$. This model is not to be confused with a speaker model. It is a model of how *all* i-vectors, for *all* speakers, are generated.

Let θ denote some hypothesis, which partitions the N elements of Φ into M speaker subsets, $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_M \subseteq \Phi$. We assume that if θ is given, $\mathcal{M}$ produces M different *speaker identity variables* (*these* are speaker models), $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M \in \mathcal{Y}$, sampled independently from $P(\mathbf{y}|\mathcal{M})$. For each speaker i, the set $\mathcal{S}_i$ of i-vectors supposedly produced by that speaker is sampled independently from $P(\boldsymbol{\phi}|\mathbf{y}_i, \mathcal{M})$, for every $\boldsymbol{\phi} \in \mathcal{S}_i$. These modelling assumptions can be represented as:

$$P(\Phi|\theta, \mathcal{M}) = \prod_{i=1}^{M} P(\mathcal{S}_i|\mathcal{M}), \tag{4}$$

$$P(\mathcal{S}_i|\mathcal{M}) = \int_{\mathcal{Y}} P(\mathcal{S}_i|\mathbf{y}, \mathcal{M}) P(\mathbf{y}|\mathcal{M})\, \mathbf{dy} \tag{5}$$

$$P(\mathcal{S}_i|\mathbf{y}, \mathcal{M}) = \prod_{\boldsymbol{\phi} \in \mathcal{S}_i} P(\boldsymbol{\phi}|\mathbf{y}, \mathcal{M}). \tag{6}$$

Notice that the speaker identity variables are integrated out in (5)—we do not need point estimates of their values in order to compute (4), the relevant likelihood for θ. The nature of the speaker model space $\mathcal{Y}$ and the details of the distributions $P(\mathbf{y}|\mathcal{M})$ and $P(\boldsymbol{\phi}|\mathbf{y}, \mathcal{M})$ depend on the generative model $\mathcal{M}$. Here we further discuss the general case, deferring the detailed description of $\mathcal{M}$ to the next section.

We proceed with the key insight that we can use the product rule to alternatively express (5) as:

$$P(\mathcal{S}_i|\mathcal{M}) = \frac{P(\mathcal{S}_i|\mathbf{y}_0, \mathcal{M}) P(\mathbf{y}_0|\mathcal{M})}{P(\mathbf{y}_0|\mathcal{S}_i, \mathcal{M})} \tag{7}$$

Notice that the LHS is independent of $\mathbf{y}_0$, so that we may choose any $\mathbf{y}_0 \in \mathcal{Y}$ to compute the RHS, as long as the denominator is non-zero.

At a first glance it may seem as if we have magically solved the integral (5), but in order to compute the normalization factor for the posterior $P(\mathbf{y}_0|\mathcal{S}_i, \mathcal{M})$, it is always necessary to integrate (at least implicitly). However, if $P(\mathbf{y}|\mathcal{M})$ is a conjugate prior [12, 13] to $P(\boldsymbol{\phi}|\mathbf{y}, \mathcal{M})$, then (7) turns out to be a convenient way to structure the calculation. This will become apparent below.

Now use (7) in (4), then expand it using (6) and simplify the nested products using the fact that the subsets $\mathcal{S}_i$ form a partition of Φ. This gives:

$$\begin{aligned} P(\Phi|\theta, \mathcal{M}) &= \prod_{i=1}^{M} \frac{P(\mathcal{S}_i|\mathbf{y}_0, \mathcal{M}) P(\mathbf{y}_0|\mathcal{M})}{P(\mathbf{y}_0|\mathcal{S}_i, \mathcal{M})} \\ &= K(\Phi) L(\theta|\Phi) \end{aligned} \tag{8}$$

where $K(\Phi) = \prod_{j=1}^{N} P(\boldsymbol{\phi}_j|\mathbf{y}_0, \mathcal{M})$ is an irrelevant data-dependent constant, which is independent of the partitioning hypothesis θ and which we need not compute when recognizing θ. The required computation is the *likelihood* $L(\theta|\Phi)$:

$$L(\theta|\Phi) = \prod_{i=1}^{M} Q(\mathcal{S}_i), \tag{9}$$

$$Q(\mathcal{S}_i) = \frac{P(\mathbf{y}_0|\mathcal{M})}{P(\mathbf{y}_0|\mathcal{S}_i, \mathcal{M})} \tag{10}$$

which we have conveniently expressed in terms of the statistic $Q(\mathcal{S}_i)$.

It turns out $Q(\mathcal{S}_i)$ is a very useful building block to put together solutions for several of the speaker recognition problems listed above. Refer to rows 2 and 3 of Table 1 and notice

[1] The name *i-vector* is mnemonic for a vector of *intermediate* size (bigger than an acoustic feature vector and smaller than a supervector), which contains most of the relevant *information* about the speaker *identity*.

that for $N > 4$, $Q(\mathcal{S}_i)$ is a more compact representation of the speaker recognition information than $L(\theta|\mathcal{S})$. The former grows as $2^N - 1$, i.e. the number of non-empty subsets of Φ, while the latter grows as B_N. However, both representations become intractable as N grows.

In section 5, we show how to compute $Q(\mathcal{S})$. Here we continue by giving solutions in terms of $Q(\mathcal{S})$, for several of the speaker recognition problems listed above:

4.1. The canonical speaker detection problem

For the canonical two-input problem, we use (9) to express the speaker detection likelihood-ratio (1) as:

$$\lambda = \frac{P(\Phi|\theta_{\text{tar}}, \mathcal{M})}{P(\Phi|\theta_{\text{non}}, \mathcal{M})} = \frac{Q(\{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2\})}{Q(\{\boldsymbol{\phi}_1\})Q(\{\boldsymbol{\phi}_2\})} \tag{11}$$

The posterior is computed with (2).

4.2. The partitioning problem

For the N-input speaker partitioning problem, the posterior for hypothesis θ is:

$$P(\theta|\Phi, \boldsymbol{\pi}, \mathcal{M}) = \frac{P(\theta|\boldsymbol{\pi})L(\theta|\Phi)}{\sum_{\theta' \in \Theta_K} P(\theta'|\boldsymbol{\pi})L(\theta'|\Phi)} \tag{12}$$

where Θ_K is the set of B_N hypotheses, and where $L(\theta|\Phi)$ is given in terms of $Q(\mathcal{S})$ by (9).

4.3. The counting problem

Here we give the solution for the counting problem with a triple-input $\Phi = \{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \boldsymbol{\phi}_3\}$. The general case is similar.

We compute the likelihood for count hypothesis θ_i in terms of the likelihoods for the associated partitioning hypotheses, by using (3) and by assuming that the partitioning hypotheses $\theta_{12|3}$, $\theta_{13|2}$ and $\theta_{1|23}$ are equally likely a-priori. The likelihoods for the three count hypotheses are:

$$\begin{aligned} L(\theta_1|\Phi) &= L(\theta_{123}|\Phi) = Q(\Phi), \\ L(\theta_2|\Phi) &= \frac{1}{3}\big(L_{12|3} + L_{1|23} + L_{13|2}\big), \\ L(\theta_3|\Phi) &= L(\theta_{1|2|3}|\Phi) = \prod_{i=1}^{3} Q(\{\boldsymbol{\phi}_i\}) \end{aligned} \tag{13}$$

where θ_i is the hypothesis that there are i speakers in Φ; and where, using (9):

$$L_{jk|\ell} = L(\theta_{jk|\ell}|\Phi) = Q(\{\boldsymbol{\phi}_j, \boldsymbol{\phi}_k\})Q(\{\boldsymbol{\phi}_\ell\}) \tag{14}$$

The posterior is:

$$P(\theta_i|\Phi, \boldsymbol{\pi}, \mathcal{M}) = \frac{P(\theta_i|\boldsymbol{\pi})L(\theta_i|\Phi)}{\sum_{j=1}^{3} P(\theta_j|\boldsymbol{\pi})L(\theta_j|\Phi)} \tag{15}$$

4.4. The extended training detection problem

Let the input set of i-vectors be $\Phi = \mathcal{T} \cup \{\boldsymbol{\phi}_t\}$, where $\mathcal{T}$ is the training set and $\boldsymbol{\phi}_t$ is the test input. Using (9), the speaker detection likelihood-ratio is:

$$\lambda = \frac{P(\Phi|\theta_{\text{tar}}, \mathcal{M})}{P(\Phi|\theta_{\text{non}}, \mathcal{M})} = \frac{Q(\Phi)}{Q(\mathcal{T})Q(\{\boldsymbol{\phi}_t\})} \tag{16}$$

The posterior is computed with (2).

4.5. The multiple-train, multiple-test detection problem

Solution (16) suggests a slightly more general solution for the case where we also have multiple test inputs known to be of the same (but unknown) speaker. Let the input set be $\Phi = \mathcal{T} \cup \mathcal{Z}$, where $\mathcal{T}$ is the training set and $\mathcal{Z}$ is the test set. Each set has one speaker, but these speakers may or may not be the same. Now the speaker detection likelihood-ratio is:

$$\lambda = \frac{P(\Phi|\theta_{\text{tar}}, \mathcal{M})}{P(\Phi|\theta_{\text{non}}, \mathcal{M})} = \frac{Q(\Phi)}{Q(\mathcal{T})Q(\mathcal{Z})} \tag{17}$$

The posterior is computed with (2).

5. The two-covariance model

Here we show how to compute (10), if we adopt for $\mathcal{M}$ a simple linear-Gaussian [12] generative model, which we call the *two-covariance model.*

The speaker model, $\mathbf{y}$, is a vector of the same dimensionality as an i-vector. We suppose that an i-vector $\boldsymbol{\phi}$ of speaker s, observed on occasion t is $\phi = \mathbf{y}_s + \mathbf{z}_t$, where $\mathbf{z}_t$ is Gaussian noise. Let

$$P(\mathbf{y}|\mathcal{M}) = \mathcal{N}(\mathbf{y}|\boldsymbol{\mu}, \mathbf{B}^{-1}), \tag{18}$$

$$P(\boldsymbol{\phi}|\mathbf{y}, \mathcal{M}) = \mathcal{N}(\boldsymbol{\phi}|\mathbf{y}, \mathbf{W}^{-1}) \tag{19}$$

where $\mathcal{N}$ denotes the normal distribution; $\boldsymbol{\mu}$ is the speaker mean; $\mathbf{B}^{-1}$ is the *between-speaker* covariance matrix; $\mathbf{W}^{-1}$ is the *within-speaker* covariance matrix; and $\mathbf{B}$ and $\mathbf{W}$ are the corresponding precision matrices. Since (18) is a conjugate prior for (19), the posterior for $\mathbf{y}$ is also normal and can be expressed [13, 12] as:

$$P(\mathbf{y}|\mathcal{S}, \mathcal{M}) = \mathcal{N}(\mathbf{y}|\mathbf{L}^{-1}\gamma, \mathbf{L}^{-1}), \tag{20}$$

$$\gamma = \mathbf{B}\boldsymbol{\mu} + \mathbf{W}\sum_{\phi \in \mathcal{S}} \phi, \tag{21}$$

$$\mathbf{L} = \mathbf{B} + n\mathbf{W}, \tag{22}$$

where n is the number of utterances in subset $\mathcal{S}$. Notice that when $\mathcal{S} = \{\}$ and $n = 0$, then we recover the prior: $P(\mathbf{y}|\{\}, \mathcal{M}) = \mathcal{N}(\mathbf{y}|\boldsymbol{\mu}, \mathbf{B}^{-1})$. For the normal posterior, it is convenient to choose $\mathbf{y}_0 = \mathbf{0}$ when computing (10):

$$\log Q(\mathcal{S}) = \frac{1}{2}(\log|\mathbf{B}| - \boldsymbol{\mu}'\mathbf{B}\boldsymbol{\mu} - \log|\mathbf{L}| + \gamma'\mathbf{L}^{-1}\gamma) \tag{23}$$

5.1. Training

The two-covariance i-vector speaker recognizer has two training steps:

1. First, the parameters of the i-vector extractor have to be trained. This is done as explained in [10], applying the EM-algorithm of [14] to a development database of multiple recordings of each of several hundreds of speakers, speaking over diverse telephone channels.
2. The same development data is re-used for the second step. We apply the newly trained i-vector extractor to map each development database recording to an i-vector. The parameters, $(\mathbf{B}, \mathbf{W}, \boldsymbol{\mu})$, of the two-covariance model $\mathcal{M}$ are then trained on this database of i-vectors. The training algorithm is another EM-algorithm [12] that maximizes the likelihood of the true

partitioning of the M speakers in this database. The EM-algorithm maximizes:

$$\prod_{i=1}^{M} P(\mathcal{S}_i|\mathcal{M}) \tag{24}$$

w.r.t. $(\mathbf{B}, \mathbf{W}, \boldsymbol{\mu})$, where $\mathcal{S}_i$ is the set of i-vectors belonging to speaker i. Our EM-algorithm was derived by regarding the speaker identity variables $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N$ as the *hidden variables*. The key to constructing the EM-algorithm is the posterior distribution for the hidden variables, given by (20).

We train two separate i-vector systems, one using male development data in both steps and the other using female data in both steps. In our experiments reported below, we apply these systems respectively to male and female evaluation data.

6. Evaluation by cross-entropy

In order to evaluate the goodness of our speaker recognizer solutions in our experiments below, we need an evaluation criterion suitable for evaluating posterior probability distributions. We consider a solution *good* if the posteriors it produces can be used to make minimum-expected-cost Bayes decisions that have lower cost on average than Bayes decisions made with the prior alone.

Let $\theta \in \{\theta_1, \theta_2, \ldots, \theta_K\}$. Let c_{ij} be the cost of the *error* when recognizing θ_j when θ_i is really true. Correct decisions have zero cost: $c_{ii} = 0$. Let the recognizer's posterior distribution for θ be $\mathbf{r} = (r_1, \ldots, r_K)$. A *user* of the recognizer would make a minimum-expected-cost Bayes decision as $k = \arg\min_j \sum_{\ell=1}^{K} r_\ell c_{\ell j}$. The *evaluator* who knows the true hypothesis to be θ_i, judges the cost of this decision as $c_i^*(\mathbf{r}) = c_{ik}$. Thus $c^*(\mathbf{r})$ forms an evaluation of the goodness of a single posterior $\mathbf{r}$.

Now let $\mathbf{r}_t = \mathcal{R}(\Phi_t, \bar{\boldsymbol{\pi}})$, $t = 1 \cdots T$ be the recognizer's posteriors calculated for the T trials of a supervised evaluation database, where $\mathcal{K}_i$ is the set of trial indices where hypothesis θ_i is really true; and where we have chosen $\bar{\boldsymbol{\pi}}$ to be uniform, so that $P(\theta|\bar{\boldsymbol{\pi}}) = \frac{1}{K}$. Then $\mathcal{R}$ can be evaluated on this database as:

$$\mathcal{C}(\mathcal{R}) = \frac{1}{K} \sum_{i=1}^{K} \frac{1}{|\mathcal{K}_i|} \sum_{t \in \mathcal{K}_i} c_i^*(\mathbf{r}_t) \tag{25}$$

This criterion is unsatisfactory in the sense that it is dependent on *fixed* values of the prior and cost coefficients. Yet, we would like it to evaluate the solution $\mathcal{R}$, which is in principle applicable to making Bayes decisions with *any* cost coefficients and any prior. We remedy this by making the cost coefficients variable and then taking the *expected value* of $\mathcal{C}(\mathcal{R})$. We do not also have to vary the prior, since $\mathcal{C}(\mathcal{R})$ is dependent only on products of cost and prior coefficients, so that varying cost is equivalent (for the purpose of evaluation) to varying cost-prior products [2, 15].

We vary cost coefficients by making them dependent on a parameter $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_K) \in \mathbb{P}_K$, so that $c_{ij} = \frac{1}{\gamma_i}, j \neq i$. This causes all coefficients (except $c_{ii} = 0$) to vary between 1 and infinity. Now representing (25) as $\mathcal{C}(\mathcal{R}, \boldsymbol{\gamma})$, and assuming a flat distribution over $\boldsymbol{\gamma}$, we define the new evaluation criterion to be proportional to:

$$\int_{\mathbb{P}_K} \mathcal{C}(\mathcal{R}, \boldsymbol{\gamma}) \, \mathbf{d}\boldsymbol{\gamma} \tag{26}$$

This integral can be solved[2] analytically [15] to give (up to an unimportant constant of proportionality) our evaluation criterion, C_{xe}:

$$C_{\text{xe}}(\mathcal{R}) = \frac{1}{K} \sum_{i=1}^{K} \frac{1}{|\mathcal{K}_i|} \sum_{t \in \mathcal{K}_i} -\log_2 r_{it} \tag{27}$$

where r_{it} is the recognizer's posterior probability for the hypothesis that is true for trial t.

This criterion can be interpreted as cross-entropy between the evaluator's and the recognizer's posteriors and has units in *bits* of Shannon's entropy [2, 15]. It takes values between 0 and ∞ as follows (θ_i is the true hypothesis for trial t):

$C_{\text{xe}} = 0$, for the *oracle* recognizer that outputs $r_{it} = 1$ for every trial.

$C_{\text{xe}} = \infty$, for a *badly calibrated* recognizer that outputs $r_{it} = 0$ for at least one trial.

$C_{\text{xe}} = \log_2 K$, for the *reference* recognizer that outputs $r_{it} = \bar{\pi}_i = \frac{1}{K}$, for every trial.

We consider a recognizer to be *good* if $C_{\text{xe}} < \log_2 K$.

6.1. Calibration

Our generative i-vector recognizer is trained with maximum likelihood as explained above. In our experiments below, we report performance of this system as is, on the counting task, but we also try a simple discriminative adaptation of the system.

We use C_{xe} as criterion to train an affine re-calibration transform of the log-likelihoods given by the two-covariance model. This training procedure is in fact just a form of logistic regression [16, 12].

Following our work in [16], we *calibrate* the counting log-likelihoods as follows. Let $\boldsymbol{\ell}_t$ be a vector of three log-likelihoods components, namely the logarithms of (13), computed for a trial with input Φ_t. Then, the re-calibrated log-likelihood vector is:

$$\tilde{\boldsymbol{\ell}}_t = \alpha \boldsymbol{\ell}_t + \boldsymbol{\beta} \tag{28}$$

where the *calibration parameters* are α, a positive scaling constant and $\boldsymbol{\beta}$, a 3-dimensional translation. When we train or apply calibration we use the exponentiated components of $\tilde{\boldsymbol{\ell}}$ in (15), in place of (13). The calibration parameters are trained discriminatively by using (15) in (27) and minimizing. Since C_{xe} is a convex function of $(\alpha, \boldsymbol{\beta})$, it has a global minimum, which can be found numerically[3] with standard convex optimization techniques [12, 17].

In order to compute C_{xe} while training calibration, one needs a supervised evaluation database. In our experiments, we report which databases were used for calibration.

6.2. Minimum cross-entropy

We define an auxiliary evaluation criterion, $C_{\text{xe}}^{\text{min}}$ as:

$$C_{\text{xe}}^{\text{min}}(\mathcal{R}) = \min_{\alpha, \boldsymbol{\beta}} C_{\text{xe}}(\mathcal{R})|_{\tilde{\boldsymbol{\ell}} = \alpha \boldsymbol{\ell} + \boldsymbol{\beta}} \tag{29}$$

[2] This is easy to show for the case $K = 2$, see [2]. Do not try this at home for $K > 2$.

[3] Our MATLAB toolkit for performing this minimization is available at `http://focaltoolkit.googlepages.com`.

This is C_{xe} for a recognizer that has undergone a 'cheating', train-on-test calibration, where the calibration has been trained on the evaluation database.

This criterion can be used to judge whether calibration that was trained on an independent database still works well on the evaluation database. Or it can be used as an indication of how well an uncalibrated system could have performed if calibration had been done.

Notice that $C_{\text{xe}}^{\text{min}} \leq \log_2 K$, because the reference recognizer is obtained at $\alpha = 0$.

7. Experimental Method

We demonstrate experimentally the performance of our two-covariance i-vector solution on a three-input problem. For convenience of exposition, we give results for the counting problem, rather than the partitioning problem. Keep in mind however, that by exercising the counting problem, we are also in effect exercising the partitioning problem, because the counting likelihoods are computed from the partitioning likelihoods via (13).

We trained the i-vector extractor and the parameters of the two-covariance model as explained in section 5.1, by using 27841 telephone conversation-sides, involving 1943 speakers from the following databases: NIST SRE 2004 evaluation data [18], NIST SRE 2005 evaluation data [19], Switchboard 2 phase 2 [20], Switchboard 2 phase 3 [21], Switchboard cellular part 1 [22], Switchboard cellular part 2 [23].

We ran the following experiments:

1. A canonical two-input detection test, the core task of NIST SRE'08.
2. A three-input counting test on NIST SRE'06 data.
3. A three-input counting test on NIST SRE'08 data, with optional calibration trained on SRE'06.

7.1. The two-input test

To demonstrate state-of-the-art performance of the two-covariance i-vector solution on a familiar task, we ran it on the telephone part of the core task of NIST SRE 2008 [6].

The detection scores were computed by using (23) in (11), followed by a symmetrized version of ZT-norm [24] for score normalization.

The system achieved EER (equal-error-rate) of 4.69% on male data and 6.71% on female data. This can be compared to the official SRE 2008 results available online [25], see the DET-curve labelled 'SHORT2-SHORT3: Telephone Speech in Training and Test'.

7.2. Three-input counting tests

The SRE'06 and '08 evaluation databases were used respectively for calibrating and testing our system on the counting problem.

Three-input trials were created by randomly selecting groups of three files from each test database in a way that produced a reasonable balance between the number of speakers per trial. Table 2 gives the number of speakers and segments available for selection during trial creation. The calls in the 2008 database were made from 2506 distinct phone numbers, so channel variability was large. Table 3 gives the resultant number of trials of each type. The fourth column gives the number of trials in which all the segments are from the same speaker and the sixth column gives the number of trials in which all the segments are from different speakers.

The *raw recognizer scores* were the count hypothesis likelihoods (13), computed by using (23). The scores were optionally calibrated as explained in section 6.1. These scores (raw or calibrated) were used to make either *soft* or *hard* decisions:

soft decisions: The scores are used in (15) to compute the recognizer's posteriors (at a flat prior of $\frac{1}{3}$). The posteriors are then evaluated by the cross-entropy criterion, C_{xe}, using (27).

hard decisions: The recognizer's estimate of the speaker count was chosen as the one with maximum posterior probability (or equivalently maximum likelihood, because of the flat prior). Hard decisions were evaluated using confusion matrix and percentage error-rate.

That is, our evaluation measures were:

percentage error rate The number of failed trials expressed as a percentage of the number of trials. In each trial, the system makes a maximum likelihood estimate of the number of speakers and this estimate is compared with the true number of speakers to determine whether the trail was successful.

cross-entropy See section 6. This is compared with $\log_2 K$ to determine whether we have built a *good* recognizer.

minimum cross-entropy See section 6.2.

calibration loss The difference between the cross-entropy and the minimum cross-entropy. This gives the performance loss for a system that has not been properly calibrated, or equivalently, the performance to be gained from properly calibrating the system.

Table 2: *Information about the testing and calibration databases.*

year	sex	# speakers	# segments
2006	m	345	1884
2006	f	340	2362
2008	m	492	1543
2008	f	844	2818

Table 3: *Trial counts for the testing and calibration databases.*

year	sex	# trials	# 1 spk	# 2 spk	# 3 spk
2006	m	900	295	290	315
2006	f	900	295	296	309
2008	m	1024	299	382	343
2008	f	2048	569	723	756

Table 4: *Results for tests on male databases.*

test	cal	C_{xe}	$C_{\text{xe}}^{\text{min}}$	cal-loss	% err
2006	-	0.92	0.24	0.68	6.67
2006	2006	0.24	0.24	0.00	5.44
2008	-	0.78	0.21	0.57	8.20
2008	2006	0.23	0.21	0.01	6.54
2008	2008	0.21	0.21	0.00	6.05

Table 5: *Confusion matrix for male 2006 data on the uncalibrated system.*

true\estm	1	2	3
1	283	11	1
2	12	270	8
3	0	28	287

Table 6: *Confusion matrix for male 2008 data on the uncalibrated system.*

true\estm	1	2	3
1	290	9	0
2	21	352	9
3	6	39	298

8. Results

Tables 4 and 8 give the results respectively for male and female three-input experiments. The columns of these tables have the following meanings:

test The database on which the test was performed.

cal The database on which the system was calibrated.

$C_{\mathbf{xe}}$ The cross-entropy for the test.

$C_{\mathbf{xe}}^{\mathbf{min}}$ The minimum cross-entropy.

cal-loss The calibration loss.

% err The percentage error rate.

The rows refer to the following test conditions:

1. Test on 2006 (uncalibrated).
2. Test on 2006 ('cheating' calibration on 2006).
3. Test on 2008 (uncalibrated).
4. Test on 2008 (calibrated on 2006).
5. Test on 2008 ('cheating' calibration on 2008).

We denote the calibrate-on-test calibrations as 'cheating', because here the true hypothesis labels are available for the system under evaluation to use for calibration. The cheating calibrations were done to see what effect ideal calibration might have.

In these tables (4 and 8), we see that calibration reduces the error rate for the 2008 male test from 8.20% to 6.54% and reduces the calibration loss from 0.57 to 0.01. The female error rate increases slightly (from 6.05% to 6.10%), but C_{xe} decreases from 0.84 to 0.24 and the calibration loss practically vanishes.

The discrepancy between error-rate and cross-entropy can be explained by noting that log-likelihood scaling has no effect on the maximum-likelihood estimates and hence no influence on the error-rate. In contrast, since C_{xe} effectively considers a wide range of operating points, it is sensitive to calibration of the log-likelihoods and is affected by both scaling and shifts. Indeed, we noticed that the main effect of the recalibration was to reduce log-likelihood magnitudes by a factor of about 10. This is to be expected, because the unrealistic and oversimplified modelling assumptions of the two-covariance model are expected to lead to overconfident likelihoods.

The reference value of C_{xe} for decisions made with the prior is $\log_2 K = \log_2 3 = 1.585$. The C_{xe} values for both the male (0.23) and female (0.24) tests on 2008 (with calibration on 2006) are well below this value, so we are justified in calling our recognizer *good*. In fact all results for the uncalibrated recognizer are also below 1.585.

Table 7: *Confusion matrix for male 2008 data on the system calibrated using male 2006 data.*

true\estm	1	2	3
1	290	9	0
2	21	344	17
3	4	16	323

Table 8: *Results for tests on female databases.*

test	cal	C_{xe}	$C_{\text{xe}}^{\text{min}}$	cal-loss	% err
2006	-	0.85	0.24	0.61	7.0
2006	2006	0.24	0.24	0.00	6.33
2008	-	0.84	0.24	0.60	6.05
2008	2006	0.24	0.24	0.0028	6.10
2008	2008	0.24	0.24	0.00	5.86

Tables 5, 6 and 7 are confusion matrices for the male tests and tables 9, 10 and 11 are confusion matrices for the female tests. The row numbers give the true number of speakers, the column numbers give the maximum likelihood estimate of the number of speakers that the system made for the trial and the matrix elements give the error counts for each combination of true count and estimated count. Entries in diagonal elements correspond to correct estimates and off-diagonal entries correspond to errors.

9. Conclusion

We propose the general speaker partitioning problem as a unification of several well-known speaker recognition tasks. We show that solving this problem in general, with a simple generative i-vector model leads to solutions of several of the more specific problems.

Our solutions are tractable for problems involving a small number of inputs, but are vulnerable to combinatorial explosion in complexity for a large number of inputs.

We show that on NIST evaluation data our generative model already works as is, but it does benefit from further discriminative calibration.

10. References

[1] Daniel Ramos-Castro, Joaquin Gonzalez-Rodriguez, and Javier Ortega-Garcia, "Likelihood ratio calibration in a transparent and testable forensic speaker recognition framework," in *Proceedings of the IEEE Odyssey 2006 Speaker and Language Recognition Workshop*, San Juan, Puerto Rico, June 2006.

[2] Niko Brümmer and Johan du Preez, "Application independent evaluation of speaker detection," *Computer Speech and Language*, vol. 20, pp. 230–275, 2006.

[3] Gian-Carlo Rota, "The number of partitions of a set," *American Mathematical Monthly*, vol. 71, no. 5, pp. 498–504, 1964.

[4] Alvin F. Martin and Craig S. Greenberg, "NIST 2008 speaker recognition evaluation: Performance across tele-

Table 9: *Confusion matrix for female 2006 data on the uncalibrated system.*

true \ estm	1	2	3
1	278	16	1
2	6	276	14
3	2	24	283

Table 10: *Confusion matrix for female 2008 data on the uncalibrated system.*

true \ estm	1	2	3
1	535	31	3
2	15	683	25
3	4	46	706

Table 11: *Confusion matrix for female 2008 data on the system calibrated using female 2006 data.*

true \ estm	1	2	3
1	545	21	3
2	20	650	53
3	6	22	728

phone and room microphone channels," in *Proceedings of Interspeech 2009*, Brighton, UK, Sept. 2009.

[5] The National Institute of Standards and Technology, "The NIST year 2006 speaker recognition evaluation plan," `http://www.itl.nist.gov/iad/mig/tests/sre/2006/sre-06_evalplan-v9.pdf`, March 2006.

[6] The National Institute of Standards and Technology, "The NIST year 2008 speaker recognition evaluation plan," `http://www.itl.nist.gov/iad/mig/tests/sre/2008/sre08_evalplan_release4.pdf`, April 2008.

[7] Douglas Reynolds, Patrick Kenny, and Fabio Castaldo, "A study of new approaches to speaker diarization," in *Proceedings of Interspeech 2009*, Brighton, UK, Sept. 2009.

[8] Patrick Kenny, Douglas Reynolds, and Fabio Castaldo, "Diarization of telephone conversations using factor analysis," submitted to IEEE Journal of Selected Topics in Signal Processing, 2009.

[9] Lukáš Burget et al., "Robust speaker recognition over varying channels," in *Johns Hopkins University CLSP Summer Workshop Report*, 2008, Online: `http://www.clsp.jhu.edu/workshops/ws08/documents/jhu_report_main.pdf`.

[10] Najim Dehak, Réda Dehak, Patrick Kenny, Niko Brümmer, Pierre Ouellet, and Pierre Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification," in *Proceedings of Interspeech 2009*, Brighton, UK, Sept. 2009.

[11] Najim Dehak, Patrick Kenny, Reda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," submitted to IEEE Transactions on Audio, Speech and Language Processing, 2010.

[12] Christopher M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer, Oct. 2007.

[13] Morris H. DeGroot, *Optimal Statistical Decisions*, McGraw-Hill, 1970.

[14] Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1435–1447, May 2007.

[15] Niko Brümmer, "Ph.d. dissertation," University of Stellenbosch, submitted March, 2010.

[16] Niko Brümmer and David A. van Leeuwen, "On calibration of language recognition scores," in *Proceedings of the IEEE Odyssey 2006 Speaker and Language Recognition Workshop*, San Juan, Puerto Rico, June 2006, pp. 1–8.

[17] Jorge Nocedal and Stephen J. Wright, *Numerical Optimization*, Springer, 2006.

[18] The National Institute of Standards and Technology, "The NIST year 2004 speaker recognition evaluation plan," `www.itl.nist.gov/iad/mig/tests/sre/2004/SRE-04_evalplan-v1a.pdf`, January 2004.

[19] The National Institute of Standards and Technology, "The NIST year 2005 speaker recognition evaluation plan," `http://www.itl.nist.gov/iad/mig/tests/sre/2005/sre-05_evalplan-v6.pdf`, March 2005.

[20] Linguistic Data Consortium, "Switchboard-2 phase II audio," `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC99S79`, 1999.

[21] Linguistic Data Consortium, "Switchboard-2 phase III audio," `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002S06`, 2002.

[22] Linguistic Data Consortium, "Switchboard cellular part 1 audio," `http://www.ldc.upenn.edu/Catalog/docs/LDC2001S15/`, 2001.

[23] Linguistic Data Consortium, "Switchboard cellular part 2 audio," `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2004S07`, 2004.

[24] Patrick Kenny, Najim Dehak, Réda Dehak, Vishwa Gupta, and Pierre Dumouchel, "The role of speaker factors in the NIST extended data task," in *Proceedings of the IEEE Odyssey Speaker and Language Recognition Workshop 2008*, Stellenbosch, South Africa, Jan. 2008.

[25] The National Institute of Standards and Technology, "The 2008 NIST speaker recognition evaluation results," `http://www.itl.nist.gov/iad/mig/tests/sre/2008/official_results/index.html`, Aug. 2008.